UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/911,819	07/24/2001	John T. Micco	04899-046001	6291

7590 07/17/2006

Kevin J. Canning, Esq.
Lahive & Cockfield, LL.P
28 State Street
Boston, MA 02109

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 07/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/911,819	MICCO ET AL.
	Examiner	Art Unit
	Tuan A. Vu	2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-56* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)	4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *7/5/06*.
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application (PTO-152)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.	6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 4/6/2006.

As indicated in Applicant's response, claims 1,3-11, 17, 21-24, 29, 31, 33-39, 49-51 have been amended; and resubmitted for prosecution are claims 1-56 in the office action.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

3. Claims 1-3, 6-10, 29-32, and 34-38 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Specifically, claim 1 recites a computing device based method having the steps of providing processing a definition of a function and creating description information therefrom. One skill in the art would not be apprised on existence of (i) a functional and concrete step acting upon the elements recited, the definition and the description information; (ii) a targeted, useful and tangible result being the consequence of an action using/operated upon, say, the *description information* thus claimed. In other words, reciting that some description *information* enables some translation of code does not convey one explicit step (of the method claim) that an action – the step of translating – is actually taken to accomplish any tangible result, when such *translation* has been construed as an a potentiality; i.e. the description information limitation thus recited is not sufficiently put in a format that would enable the construction of an action leading to a practical application, i.e. leading to an useful tangible result. The disclosure teaches about translation program call from one language to another, and if the translation of code is the useful and practical application using a computing device, that form of useful application is not clearly stated in the claim. The element 'description information' as recited fails to reasonably the convey that a actual step is taken in order to generate a useful and tangible result, making the use of the element a practical application. And as such, the claim does not amount to a practical application because it fails to yield a concrete, useful and tangible result as required by the above-mentioned Practical Application Test. The claim is rejected for leading to a non-statutory subject matter.

Dependent claims 2-3 and 6-10 for not remedying to the abstract idea deficiency of the base claim are also rejected.

Claim 29 recites computer-readable instructions to obtain a definition in a first language and create description information. As set forth above, the so-called *description information* being recited in a format as to 'enables a translation' (i.e. *enables* being interpreted as a mere potentiality of something that is not really materialized into practical/useful realization) does not sufficiently convey that an action --by the program instructions-- is taken so that it would perform a real world application result, e.g. translation of a computer function call. As a whole, the claim fails to lead to a practical application whereby a concrete, useful and tangible outcome is generated. Claim 29 amounts to a non-practical application because it fails to yield a concrete, useful and tangible result as required by the above-mentioned Practical Application Test. The claim is rejected for leading to a non-statutory subject matter.

Dependent claims 30-32 and 34-38 for not remedying to the abstract idea deficiency of the base claim are also rejected.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-7, 11-13, 17, 20, 29-35, 39, 40-41, 45, and 48 are rejected under 35 U.S.C. 102(b) as being anticipated by Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering, Vol. 15, No. 11 (hereinafter Shannon).

As per claim 1, Shannon discloses a method comprising

providing a definition of a function associated with the first language about a function in a first language (e.g. … *existing languages … LISP*, *Diana structures … mapped …to C, Ada Breadboard Compiler* – Introduction pg. 1333-1334; …*data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration … function => name; String; parameters: Seq of … formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335; *Process declaration* - Fig. 2);

creating description information or *IDL* (*IDL specifications* - Fig. 3, pg. 1336; *IDL specification … specifically, C macros … and functions* – pg. 1334, left column, 4th para – Note:

The conversion based on a special package extending existing language structures to target C or Ada reads on creating a description interface on structures, e.g. a class definition, in first language, - into Ada function or C data structures; *data structures found in procedural programming languages* – 2nd para, right column, pg. 1334 – Note: the use of existing programming constructs reads on function in first language when data structure is inherent to function of a language),

wherein the description information enables translation of a call to the function into a call to a corresponding function in a second language (*…should be permitted by the C compiler* - Introduction pg. 1333-1334) without requiring processing of the definition of the function (e.g. … *should be natural and not require knowledge of implementation details* – pg. 1333, right column).

As per claims 2 and 4, Shannon discloses a file of description items and derived description information (e.g. sections II, III – pg. 1334-1339).

As per claim 3, Shannon discloses

examining the definition of the function associated with the first language (e.g. …*data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration … function => name; String; parameters: Seq of … formal parameters*, pg. 1334, right column – Note: the fact of creating appropriate parameters for a function disclose examining definition of a function in a source language);

and derive information (*IDL specifications* - Fig. 3, pg. 1336; *IDL specification … specifically, C macros … and functions* – pg. 1334, left column, 4th para – Note: generating of

IDL from existing functions written in some languages inherently teaches derive information for said functions) about the function.

As per claim 5, Shannon discloses creation of C language constructs, hence has implicitly disclose the *.lib* files associated with the assembling of object files prior to linking in C. Therefore, Shannon has implicitly disclosed library wherein entries associated with the assembling process in C compiler because at the time the invention was made it was a known concept that C compiler create lib files during a pre-linking compiler process.

As per claims 6 and 7, Shannon discloses a declaration with a set of input and output port (ch. B – pg. 1335; Fig. 2, pg. 1336) and input/output mapping (pg. 1338, Private types - right column); hence has disclosed analysis of the first language so to derive input/output formal parameters leading to creation of C formal parameters, i.e. list of input or output/return parameters otherwise the target function declaration would not result in a correct function declaration (Note: the declaration of formal input and output, and scope of variables declared in a function in 4th generation like C was a known concept at the time the invention was made; and according to C code translation by Shannon, this reads on input/outputs as in a formal declaration of a function as set forth in claims 1, 3).

As per claim 11, Shannon discloses a method in a computing device, comprising providing a file of description items, each including information about a function in a first language (e.g. *Diana structures ... mapped ...to C, Ada Breadboard Compiler* – Introduction pg. 1333-1334; *...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335), wherein the description enables translation of a call to the function in a first language into a call to a corresponding function in a second language (*...should be permitted by the C compiler* - Introduction pg. 1333-1334) without requiring processing of the definition of the function (e.g. ... *should be natural and not require knowledge of implementation details* – pg. 1333, right column); and

using information about a function associated with the first language to translate a first program file into a second program file (*...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335 – Note: using existing programming language source to analyze structures in order to generate IDL constructs reads on using information about function in 1st language to translate a first program into a second program file – see pg. 1344 Runtime efficiency: *mapping, compiler*).

As per claims 12-13, referring to rationale of claims 6-7, Shannon has disclosed analysis of the first language input/output formal parameters leading to creation of C formal parameters, i.e. list of input or output/return parameters otherwise the target function declaration would not result in a correct function declaration (Note: the declaration of formal input and output, and scope of variables declared in a function in 4th generation like C was a known concept at the time the invention was made; and according to C code translation by Shannon, this reads on input/outputs as in a formal declaration of a function as set forth in claims 1, 3).

As per claim 17, Shannon teaches using existing programming language source to analyze structures in order to generate IDL constructs reads on using information about function in 1st language to translate a first program into a second program file – see pg. 1344 Runtime efficiency: *mapping, compiler)*; hence discloses for each call in the 1st program file, retrieving an item from the file of description items, and using information description in the item to translate the first language function into a translated call corresponding to the 2nd language (Note: IDL and call declaration to functions by Shannon – refer to claim 11 -- read on retrieving item from description items; and using this information to translate into a corresponding function call in another programming language, a function call being translated reading on being stored after compiler translation).

As per claim 20, refer to rationale as set forth in claims 12-13.

As per claim 29, this is the computer-medium product version of claim 1, hence is rejected with the corresponding rejection as set forth therein.

As per claims 30-35, these are the computer product claims corresponding to claims 12-13; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claim 39, this is the computer-medium product version of claim 11, hence is rejected with the corresponding rejection as set forth therein.

As per claims 40-41, these are the computer product claims corresponding to claims 12-13; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claim 45, this is the computer-medium product version of claim 17, hence is rejected with the corresponding rejection as set forth therein.

As per claim 48, refer to rationale as set forth in claims 12-13.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 8-10, 14-16, 18-19, 36-38, 42-44 and 46-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering, Vol. 15, No. 11, in view of Bjarne Stroustrup, "the C++ Programming Language", 2nd Edition, copyright 1991 (hereinafter Stroustrup).

As per claim 8, Namespace and function scope involving local and global parameters are known in 4th generation like C. Stroustrup discloses C programming language and namespace for addressing scope of members of a function dictated within a declaration directives of such namespace (see pg. 634, chp. 3.3.1). Hence, it would have been obvious for one skill in the art at the time the invention was made in light of Shannon type checking and preprocessing for runtime efficiency analyze a function scope while creating of description information about a function of the first language in order to derive a correct scope when effecting a C function declaration according to Stroustrup; because this would support the scope of definition of parameters by Shannon's C directives because this helps support scope of members within a class or function definition as endeavored by Shannon's preprocessing macro directives (see Shannon: pg. 1341-1344).

As per claims 9 and 10, Shannon does not explicitly teach determining of variable arguments in a function and a variable return of results. But according to Stroustrup's teaching of C++ providing a variable number of arguments, e.g. input arglist[], or argv[]; or a variable output/return in form of an array, or pointer to a struct or linked list (Stroustrup: *argv[]*– chp. 3.1.6, pg. 87; pg. 485 chp. 3.4; *argument ... list* - chp. 8.2.5 – pg. 532), this a known concept C construction at the time the invention was made. Hence, for one skill in the art at the time the invention was made, determining whether a function to have a variable input arguments or return variables would also have been obvious according the above teaching by Stroustrup to address possibility where number of arguments can vary, and analyzing and precisely controlling on such extensibility and variability of parameters as endeavored by Shannon' s type and code checking for runtime as set forth above would allow the target code to accommodate for such eventuality, using the known approach provided by C as mentioned above.

As per claims 14-16, refer to the rationale of claims 8-10.

As per claims 18-19, these claims subject matter fall under the ambit of the subject matter of claims 15-16; and are rejected using to rationale as set forth in claims 15-16, respectively.

As per claims 36-38, these are the computer product claims corresponding to claims 14-16; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claims 42-44, these are the computer product claims corresponding to claims 14-16; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claims 46-47, these claims correspond to the subject matter of claims 15-16; and are rejected using to rationale as set forth in claims 15-16, respectively.

8. Claims 21-28, and 49-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Elmroth et al., "A Web Computing Environment for the SLICOT Library", December 2000, Brite-Euram III, Networks Programme NICONET, in view of Research Systems, "IDL", copyright 1994, further in view of Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering.

As per claim 21, Elmroth discloses a method in a computing device comprising:

providing a library file including functions defined by a first language (e.g. *SLICOT Library, BLAS, LAPACK* – pg. 1, Introduction; *Riccati equations* - Fig. 2-3; *...uploaded ... Matlab files, data files, Latex, Scilab* – pg. 2, pg. 6, Fig. 1, 4 – Note: uploaded matrices in Matlab or Latex format , or Riccati equations read on library files - i.e. files served as input to a library generating tool - defined in one language, all such file integrated into the SLICOT library file framework);

creating a function library (e.g. *SLICOT Routines* – Fig. 6) and a description file (e.g. PHP scripts, Aux Routines – pg. 6-7 – Note: php scripts reads on being derived via parsing <u>from</u> the SLICOT and Aux Routines), the function library including one or more functions defined by a second language, each function in the function library being translated version of a function in the library file (e.g. *SLICOT routines* – pg. 6; *...written in C or Fortran* – pg.8: Conclusions and Future Work), and

using the description file to translate a program file from the first language into the second language (e.g. *SLICOT routines, Matlab binaries, PHP scripts* – pg. 7-8 – Note: library files xxxxMD or xxxOD files in conjunction with PHP scripts and converting math functions into m-files read on one or more functions translated from the library file in a second language, i.e. Matlab binaries function files)

But Elmroth does not explicitly disclose the description file including description information that enables translation of a call to the function in a first language into a call to a corresponding function in a second language without requiring processing of the definition of the function; nor does Elmroth disclose that each call in the program file to a function in a first language is translated into a call to a corresponding function in the second language. Converting math functions into another program like high-level programming language like Fortran or C (analogous to suggestion by Elmroth – pg. 8: Conclusions and Future Work) was a known concept at the time the invention was made. Research_Systems, as set forth in claim 1, discloses definition language (IDL) and mapping various application functions to a fourth-generation programming language like C, and using such IDL file, i.e. description file similar to PHP scripts by Elmroth, to implement applications similar to the Riccati Equations or Matlab files by Elmroth, e.g. Math functions, graph plotting, Image Processing (see Research_Systems, pg. 1-5). The high-level definition description file by Research_Systems was an interactive definition language (IDL) well known at the time the invention was made for converting functions in one language into a corresponding functions in another language, like from Math functions to C program as taught by Research_Systems. As set forth in claim 1, Shannon in a similar approach as Research_Systems, also discloses mapping functions from one language to syntax and constructs implemented in C functions (re claim 1). Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide an IDL as taught by Reseach_Systems and Shannon to enhance the scripts by Elmroth so that the IDL description information is used instead of the PHP scripts, the IDL as to enable translation of a call to the function into a call to a corresponding function in a second language without requiring processing of the definition of the function; such that each call in the first language program file to a function in the library file is translated into a call to a corresponding function in the second language (re claim 1: Shannon). The benefits of using IDL would have been obvious because of the same reasons listed by Shannon: the IDL provides type safe implementation into a high-level programming like C, and further does not require processing of the definition of the function (see Shannon: Introduction, pg. 1333-1334).

As per claim 22, this claim includes the limitation as to translate a call to the function in a first language into a call to a corresponding function in a second language as in claim 21; hence is rejected as set forth therein. Further, Elmroth discloses a translated version of each function in the library file (*SLICOT Library, BLAS, LAPACK* – pg. 1, Introduction; *Riccati equations* - Fig. 2-3; *...uploaded ... Matlab files, data files, Latex, Scilab* – pg. 2, pg. 6, Fig. 1, 4- Note: uploaded matrices in Matlab or Latex format , or Riccati equations read on functions in the library file - i.e. files served as input to a library generating tool - defined in one language, all such file integrated into the SLICOT library file framework).

As per claim 23, this claim limitation corresponds to that of claim 3; hence is (in view of the combination Elmroth and Research_Systems and Shannon from above) rejected as obvious using most of Shannon teachings, mapping of function using a description file derived from examining a function in the first language library file, in light of Research_Systems.

As per claims 24 and 25, these claims correspond to limitations of claim 3, 4 and 11; hence are rejected using the corresponding Shannon's teachings in view of the rationale to combine Elmroth, Research_Systems and Shannon as set forth above.

As per claim 26-28, Elmroth discloses a Web call mapping and converting interface (Fig. 6), while Shannon discloses an interface to check call for error and type violation (the User Interface – pg. 1344). In light of the rationale as to combine the IDL teachings by Shannon with Elmroth's web interface and PHP scripts, the motivation to provide Elmroth' s Web interface a function evaluation interface as suggested by Shannon would have been for the same reasons as combining Elmroth with Shannon as in claim 21. Further, Elmroth does not specify variable input descriptor, variable output descriptor, descriptor for a known number of input or output arguments as recited in claims 18-20. In view of the rationale to combine Elmroth with the IDL by Research_Systems and Shannon, these claims will be rejected as in claims 18-20 respectively.

As per claim 49, this is the computer-medium product version of claim 21, hence is rejected with the corresponding rejection as set forth therein.

As per claims 50-56, these are the computer product claims corresponding to claims 22-28; hence are rejected with the corresponding rejections as set forth therein respectively.

Response to Arguments

9. Applicant's arguments filed 4/6/2006 have been fully considered but they are not persuasive. Following are the Examiner's corresponding counterpoints.

Rejections under USC §101:

(A) It has been argued (Appl. Rmrks, pg. 13, top) that by reciting 'enables ... the translation' along with the 'description information' as claimed, the claims would be sufficient to make the recited 'description information' a tangible useful result, a result for a translating step that is further supported by the claims 11, 21, 39 and 49. The rejection has now pointed out that the claim, by presenting a information in a format that does not sufficiently convey an actual step taken in order to provide a result, would still does not conform to a invention that is otherwise construed as a practical application. A piece of information – a description information stored in a computer -- claimed as having the ability of doing something cannot be equated to a step action taken as one would expect in one method claim like that of claim 1 or 29. The analogy is a scenario in which a method claim recites a program code that is configured to enable a transformation: the program code with a potential use fails to set forth a method step which is taken to provide a practical use leading to a tangible result.

Rejections under 35 USC §102:

(B) Applicants have submitted that the Examiner is incorrectly confusing the translation of a a function with translation of a call to a function to a call to a corresponding function(Appl. Rmrks, pg. 14, bottom) and Shannon does not appear to have taught translation into a call 'without requiring processing of the definition of the function'. It is noted that the IDL tool is to provide all the necessary description information enabling a target language programming constructs to be converted therefrom. Because the claim does not recite more specifics as to enable one skill in the art to acquire adequate understanding behind the concept of 'translating a call' into a call (to a corresponding function), such call translation has been interpreted broadly as a translation of a function for a runtime using a IDL definition by Shannon, i.e. effect code constructs in a another programming language in Shannon runtime. Thus the linking as set forth in the rejection reads on function calls translated from IDL definition based on a first language constructs (see Shannon: *...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335); hence Shannon has disclosed such translation of a call via translation of a function for a runtime using said IDL definition of a function call. The limitation recited as 'without requiring processing ... of the function' has been interpreted as reasonably broad as Shannon's IDL teaching has allowed. Since Shannon's IDL has established all the needed information of how implementation of one (program) function would require in order to facilitate the translation thereof into a corresponding function written in a target language, the interpretation is that the above limitation amounts to just using the IDL to effect such translation, i.e. no need is required to fetch the original first language expression of the function when the presence of the IDL is just to obviate just that. As set forth above, the claim is not providing specifics as to what this translation really consists of; nor the 'without requiring processing' limitation amounts to (i.e. a limitation has to lay out what is done to achieve a desired objective, not what is not needed to achieve so); nor does the claim clearly provide a executed translation step to effectuate this requirement. Since the claim language is deficient in specific teaching, Applicants in view of the above arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(C) Applicants submitted that Examiner used Shannon's separate teachings and allegedly thought to have fulfilled the claim by mingling IDL with the first language and description information (Appl. Rmrks, pg. 15, top). This argument is not taking under consideration the detailed mapping effected in the rejection wherein the Examiner has pointed out parts that represent function in a first language, a description information, and what is second language translated function.

Rejection under 35 USC §103(a):

(D) As to arguments on claims 8-10,14-16,18-19,36-38, 42-44, and 46-47, Applicants have submitted that there is no teaching or appropriate anticipation in Shannon for each and every element of the claim(s) (Appl. Rmrks, pg. 15, bottom) in order for the rejection by adding

Stroustrup to overcome the deficiencies of Shannon. This argument is not persuasive because of the rebut set forth in section B above.

For the above observations, the rejection will stand as set forth above.

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

[illegible]

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

VAT
July 7, 2006